Computers in Support of Musical Expression

Bryan-Kinns, Nick

ISSN 1470-5559

Computers in Support of Musical Expression

Nick Bryan-Kinns

Queen Mary
University of London
Department of Computer Science

RR-04-03

November 2004

Computers in Support of Musical Expression

N. BRYAN-KINNS
Interaction, Media, and Communication Research Group,
Department of Computer Science,
Queen Mary, University of London

November 9, 2004

Abstract

Music creation and performance are basic and distinctive forms of human creativity which constitute an important application domain for computer systems. In this paper we review recent developments in the use of computers to support musical expression in terms of two characteristics: level of intervention in the music produced, and level of support for collaboration between musicians. These characteristics provide a mechanism to lay out the space of current research, to identify trends, and to speculate on future research directions. Key areas for future research are identified, in particular, the need for design guidelines to inform instrument development, development of instruments which are easy to learn yet provide depth of expression, and increased support for collaboration within shared instruments.

Keywords Audiovisual, Collaboration, Creativity, Human computer interaction, Music, Sound, User interfaces

1 Music creation and performance

Collaborative improvisation, composition, and performance of music constitute a basic and distinctive form of human interaction. Titon (1996) provides ethnomusicological discussion of music and improvisation from rain forest cultures such as the BaAka people in which rich polyphonic music is socially improvised in a community setting, to classical Indian music where performances are a balance between precomposed and improvised music, and on to contemporary blues improvisation. Moreover, the separation between audience and performer(s), and between composition and performance, typical of western 'art music' are not representative of musical performance in general. Across cultures, the production and enjoyment of music is typically an open, collaborative, and 'ubiquitous' (Sloboda & O'Neill, 2001). These features make music production an important application area for computer systems, and moreover, the requirements of such behaviour pushes our development of systems in innovative and rewarding directions. We can use computers as musical instruments, as composition tools, or even musical partners in a piece. Before delving in to the details of this area some terminological issues need too be discussed.

The definition of terms such as music, improvisation, instrument, performance, and composition is an ongoing philosophical debate. Whilst some see improvisation as a completely ad-hoc activity lacking in design and method cf. (Bailey, 1992), others see it as a complex musical activity in which the creative mind is on display (Titon, 1996). Similarly, trying to differentiate between the composition of music, its performance, and improvisation is problematic cf. (Bowers, 2002), let alone determining what the difference between music and sound is. For example, Russolo (1913 1986) introduced the use of everyday sounds into futurist music which for many at the time would simply have been a cacophonous noise, much as many people may regard Steve Reich's music today.

For the purposes of this survey, which concerns itself with user interfaces to computers, we use the following definitions to ease our discussion. First, music is the socially constructed interpretation of sound cf. (Sloboda & O'Neill, 2001); so, our computer interfaces produce sound which we hope will be interpreted as music. Second, we draw no distinction between composition, performance, or improvisation as activities; we are interested in what the requirements on computer systems are such as being able to store, replay, and manipulate sounds. Third, we consider any object used to produce sound an instrument.

1.1 A Brief Recent History of Music and Technology

We could trace music instrument production back to the use of sticks to create rhythm, or reeds to create tones. Here we trace a much shorter history of mechanical musical instruments which directly preceded computer based sound production. Early forms of devices for storing and reproducing music were mechanical devices such as music boxes, orplayer pianos which played pre-composed pieces from the 1880s onwards (see (Ord-Hume, 1980)). These played longer loops of music than the barrel style devices which had been in use for bell ringing for many hundreds of years previous (Bowles, 1970). Notes to be played are indicated by pins or holes in a rotating disc which caused plucking of musical combs or striking of percussive instruments such as drums. In a typical music box such as a Polyphon, the disc rotated whilst the play-head with associated instruments remained static underneath it. Figure 1 illustrates such a device with a disc of holes indicating notes and a musical comb underneath (illustrated in grey). Some interesting early patents for disc-playing musical boxes included a re-pinnable disc in 1882 which allowed consumers as well as producers to create and edit the stored music. There are some striking similarities between such mechanical devices and the computer based instruments developed over a hundred years later and laid out in this paper.

Following on from physical production systems, electronics have been used to produce audio as far back as 1897 and the invention of the Dynamophone which could produce a wide range of pitched sounds of different timbres (Manning, 1985). Many developments followed including the emergence of electronic instruments in the 1920s and 30s such as the electric guitar which uses magnetic pick-ups to capture the vibrations of the metal strings which can then be amplified and processed in different ways, and the Thrmin which uses changes to magnetic fields in the horizontal and vertical plane caused by the position of a player's hands to determine the pitch and volume of a synthesised note. These two instruments illustrate a key distinguishing feature of innovative musical devices - some attempt to replicate or augment conventional acoustic instruments

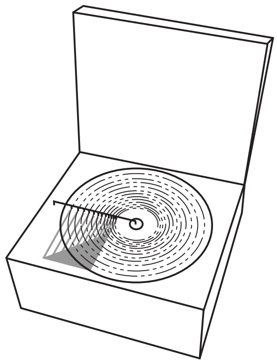

Figure 1: Polyphon

as closely as possible, whilst others attempt to develop new aesthetics of sound (Crowhurst, 1971). The latter epitomises Russolo's 1913 Futurist manifesto on The Art of Noises (Russolo, 1913 1986) which advocates the use of everyday and mechanically produced sound in musical performance and is summed up by the following sentiment:

> For an electronic instrument to be musical, it must produce sound that, whether is resembles traditional musical instruments or not, is acceptable as providing a legitimate musical form, capable of being applied to musical composition or rendition. Opinions may sometimes differ as to whether the results achieved is always musical, but in its most basic sense, this premise is sound
>
> (Crowhurst, 1971)

The second stage of electronic music production was the development of electronic audio recording equipment such as Stille's early work on recording audio onto magnetic tape in the 1920s, and the development of more robust tape recorders in the 1940s (Holmes, 1985). Such devices allowed for the recording, and more importantly, electronic manipulation of sound for later playback, and were typically to be found in electronic music studios often sponsored by national radio stations (Ernst, 1977). Finally, the third stage of electronic music development can be considered to start with the introduction of the electronic synthesizer by Moog in 1964 (Holmes, 1985) which allowed players to process, modulate, and mix any sound. Such synthesizers are analogue instruments - they manipulate continuous electronic signals such as sinusoidal waves which in turn drive speakers to produce auditory output. Development of newer forms of musical interaction have relied on digital technology such as the computer systems discussed in this paper - they manipulate numeric representations of sound which at some point must be converted into analogue signals in order to produce sound through a speaker.

The use of computers to produce audio had to wait until they had sufficient speed and processing power. Early developments included Mathews' MUSIC I to V developed at Bell Labs in the late 1950s and through the 1960s (Matthews, 1969); see (Roads, 1980) for an interview on the development) which was capable of producing very simple sounds based on four triangle-wave functions.

In 1985 Pennycook surveyed the field of computer-music interfaces (Pennycook, 1985) and Roads surveyed the field of research in music and artificial intelligence (Roads, 1985). These captured the field of new developments at the time and included consideration of the user interface issues, how signals could be processed, theories of music, as well as AI theories and how they could be applied to music generation. Looking back on both Pennycook's and Roads' reviews it is clear that interaction is a key element of music only superficially covered by the systems discussed. Also, there is a clear distinction drawn between composition and performance leaving little scope for improvisation, and moreover, interaction with algorithms as instruments, or with music as a group activity is not touched on. Since 1985 computers have continued to follow Moore's law cf. (Moore, 1965) and have now become so compact and affordable that systems mentioned in Roads' review may now be feasible on mobile telephones which integrate sufficient computing power with polyphonic audio production. This survey seeks to elucidate current developments in computer support for music focussing on the interactive aspects of the systems.

2 Current research

Interfaces can be used and subverted in a number of ways. In this paper we lay out the space of musical interaction in terms of explicit characteristics of systems rather than their intended use given the problematic definitions discussed in the introduction. For example, a traditional violin can be used in both performance and composition, but in itself it has no explicit mechanisms for storing or editing compositions, so we characterise it as an instrument purely for sound production. Similarly, a conventional piano has no explicit mechanisms to support collaborative play by two or more players so it is characterised as an individual instrument whereas, as is discussed later, some systems have explicit support for collaborative music making as part of their design.

2.1 Decomposing support

Fundamentally, music technology transforms our physical actions into sound; it somehow intervenes in music production in order to expand the range of sounds we can produce beyond those of our own bodies. Our tools have developed to such an extent that they can play with us, not just responding to direct control from us. Characterising the transformation of physical action to sound as a deterministic mapping does not sufficiently capture the range of transformations that are possible cf. (Hunt, Wanderley, & Paradis, 2002)(Wessel & Wright, 2001); 'mapping is a less useful concept when applied to the structure of complex and interactive instruments in which algorithms generate control information' (Chadabe, 2002). Indeed, it is argued that even conventional instruments such as the clarinet do not provide a direct mapping from player input to sound output, but rather, a complex combinatorial mapping of input such as breath and lip pressure as well as key presses (Rovan, Wanderley, Dubnov, & Depalle, 1997). Instead we view the transformation from physical action to music production on a dimension of determinism of intervention from simple deterministic mapping of control to sound production to non-deterministic collaboration with a system. Between these two extremes we find systems which allow us to string

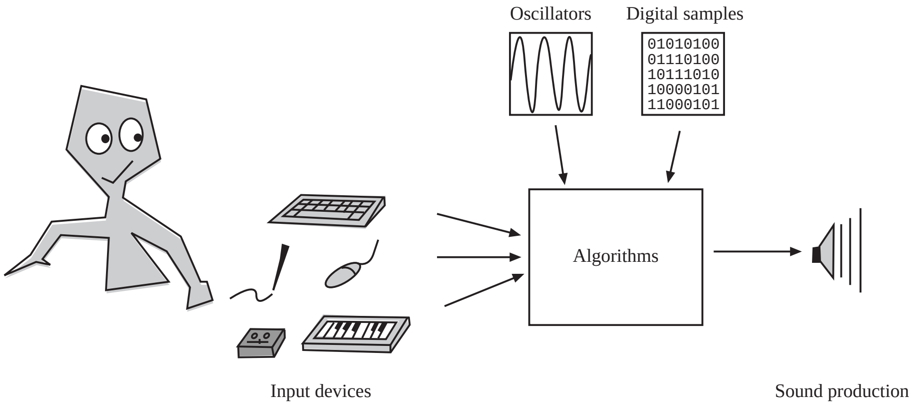

Figure 2: Transformation from user input to sound output

together and edit musical sequences and those which become less deterministic by somehow changing the music we make by, for example, correcting timing. A similar decomposition is proposed by Li (1999) with regard to the role of computers in composition and the philosophical question of 'who or what is making the music' when computers are involved. Figure 2 illustrates the basic interaction in our model; users provide some input through a variety of input devices ranging from mice and pens to keyboards and boxes with various input elements. These inputs are then used by algorithms in a range of ways discussed in this paper to produce sound from a range of sources such as oscillators for sine wave generation and digital samples.

Music is fundamentally a social process cf. (Sawyer, 2003) so our secondary dimension is the level of explicit support for collaboration between musicians. This starts with no explicit support as exemplified by traditional instruments where collaboration is co-ordinated by musicians through external representations such as musical scores, human interaction through gestures, and intonation in the music itself. At the other extreme of this dimension we see systems which have inbuilt support for collaboration such as allowing musicians to share and edit each others' compositions, or co-ordinating locally produced music. We can further decompose support for collaboration into explicit support for co-present or remote collaboration. Co-present collaboration occurs when performers are physically near each other, e.g. in the same room, and so could communicate between themselves without any technological support. Remote collaboration completely relies on technology such as text chat or telephones to support communication between performers. Of course, there may be situations in which the players are physically close enough to be able to communicate with each other, e.g. on a stage, but the physical setup is such that normal communication is not possible e.g. soundproof partitions separate players. In such situations we regard the collaboration as remote even though they are physically close as they are unable to communicate without any assistance. Blaine and Fels (2003) survey 17 current interfaces which support collaboration in some way. In this paper we add to this cannon and also draw out the level of determinism of mapping supported in order to situate collaborative with non-collaborative in-

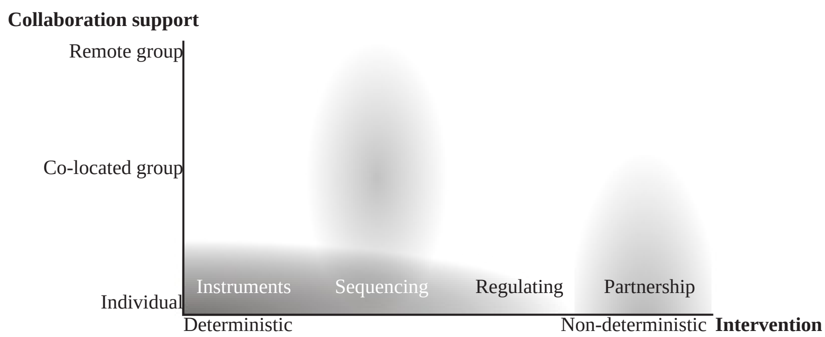

Figure 3: Two dimensions of musical interfaces

terfaces. Figure 3 illustrates our two dimensions of intervention and support for collaboration which are decomposed into the categories we discuss in this paper. Furthermore, shading on the diagram provides an indication of the amount of work in that area highlighting the predominance of work on individual instruments, and the relatively sparse work on support for collaboration. We examine systems primarily along the intervention dimension, and at each stage considering the explicit support for collaboration. The different characteristics are not discrete points along the dimension, but rather overlapping spaces which provide us with a rough means of distinguishing and comparing systems.

2.2 A note on underlying technologies

In this paper we are not concerned with the underlying tools used to construct musical interfaces per se (though clearly tools constrain what can be constructed), rather we are interested in the kinds of computer based interaction that has, and could be developed given our conceptual framework. There are many tools and systems which support the development of real-time computer based systems which transform various input controls to audio and visual output, for example Pure Data (Puckette, 1996). Pure Data allows users to construct complex data interconnections between inputs, user defined data processors, and outputs such as audio or video for real time performance and interaction. Using a graphical patch based metaphor the approach attempts to provide easy access to complex underlying code such as digital signal processing. Many of the systems surveyed in this paper use PD or a similar approach to construct their interaction. On a more physical level, there have been several attempts to modularise the physical building blocks of computer based instruments. For example, Bongers and Harris (2002) developed a set of physical input devices referred to as 'instrumentlets' which had various degrees of freedom, range, and physicality. The intention behind such approaches is to make the construction of the physical input to computer music instruments easier, but it is interesting to note that most physical input devices are still constructed from scratch on an ad-hoc basis.

Underlying all computer music devices is some sort of music exchange protocol for communicating information between various parts of the system such as the processor and the sound production card. By far the most prevalent technology used is MIDI developed in 1983 (Musical Instrument Digital Interface; (MIDI, 1996)). MIDI encodes data between controllers and instruments. Nowadays the controllers take a wide range of forms from conventional piano style to computer based play back of pre-recorded MIDI files. There are some problems with using MIDI to communicate between novel devices such as its in built assumptions about the structure of the synthesizer (based on a series of soundbanks) and its assumptions about timing mechanisms. Recent developments such as Open Sound Control (Wright, Freed, & Momeni, 2003) have started to address these issues and develop a protocol more suited to rich computer based interaction and interconnection. However, given the pervasive nature of MIDI in audio devices it is likely that it will remain dominant for the foreseeable future.

3 Instruments - Deterministic Mapping

Probably the most intuitive form of intervention in the music production process is the direct mapping of player input to audio signal production as characterised by musical instruments. Indeed, most new interfaces for musical production provide deterministic mapping from input to sound - this section provides a characterisation of the most pertinent examples of such devices. Conventionally this mapping would be achieved by some physical action such as plucking a string on a guitar causing sound to be produced. This section lays out typical current developments in computer based support for music production where there is a direct mapping between player input and audio output. Designing such instruments could draw on both principles of conventional instrument construction, and principles of human-computer-interaction as outlined in Orio et al. (2001). Such principles consider issues such as the speed of moving a selection device such as a mouse to an input region such as a button on a display. Moreover, Orio et al. provide a useful characterisation of the kinds of musical production tasks that user interfaces will need to be able to support for music production. These tasks are outlined below and are worth bearing in mind when considering the interfaces described in the rest of this section.

> Isolated tones, from simple triggering to varying characteristics of pitch, loudness, and timbre;
>
> Basic musical gestures: glissandi, trills, grace notes, and so on;
>
> Simple scales and arpeggios at different speed, range, and articulation;
>
> Phrases with different contours, from monotonic to random;
>
> Continuous feature modulation (e.g. timbre, amplitude or pitch) both for a given note and inside a phrase;
>
> Simple rhythms at different speeds combining tones
>
> (Orio, Schnell, & Wanderley, 2001).

In this section the field is first categorised in terms of explicit support for collaboration - instruments intended for individuals followed by instruments with explicit support for collaboration. Instruments intended for individuals are further categorised with reference to their physical relationship to conventional musical instrument forms (those which have previously not required any

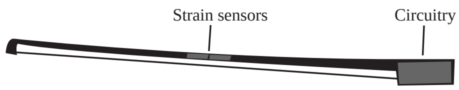

Figure 4: Hyperbow

computer support in order to produce audio signals) - first augmented versions of conventional instruments are discussed, then virtual versions of conventional instruments, and finally new forms of musical instrument. Instruments with explicit support for collaboration are categorised in terms of location - whether they are intended to be co-located, or remote.

3.1 Individual Augmented Instruments

An augmented instrument adds controls and sensors to a conventional musical instrument in order to so somehow change the deterministic sound production. A typical augmented instrument, not to be confused with the generic term hyper-instrument (Machover & Chung, 1989), is the hyperbow controller (Young, 2002). This device measures changes in the 'position, acceleration, and the downward and lateral strains of the bow' using a range of sensors attached to a conventional violin bow as an electric violin is played. Figure 4 illustrates the additions to a conventional bow - strain sensors send data to circuitry on the frog which includes accelerometers and wireless connectivity to transmit data which is then typically used to control transformations of the sound signal produced by the violin rather than directly acting as a source of musical input themselves. So, there is a mapping from the additional sensors to the input provided by the conventional instrument (the electric violin in this case).

A more extreme form of augmentation is to take the conventional form-factor of an instrument and replicate its sound production using sound models and measuring all player input such as fingering and breath. Instruments such as the EpipE (Cannon, Hughes, & Modhráin, 2003) and the HyperPuja (Young & Essl, 2003) follow this approach. The EpipE aims to capture all fingering of a Irish Uilleann Pipes using high fidelity sensors on the tone holes. Fingering information is then transformed into audio using synthesis models of Uilleann pipes. In such a situation the designers are attempting to replicate a conventional instrument through sensing, modelling, and synthesis, but of course, the high fidelity input captured by such devices could be transformed in other ways to provide new forms of audio. We consider them augmented instruments as they take the conventional physical form factors and use technology to reproduce (or enhance) their audio production.

Finally, some approaches take a conventional instrument as an input device and augment it with an additional controller. Mandellis (2002) takes this approach with the Genophone which augments a conventional piano style keyboard with a dataglove to manipulate parameters of the synthesizer. They conducted tests of the approach in which they found that users were quickly able to grasp the use of the glove to manipulate the parameters of the sound being produced in combination with the keyboard. This may help to make construction of new forms of sounds more accessible to people, and it would be interesting to explore the use of another dataglove for pitch control instead of the conventional keyboard. A finer grain of control is exemplified by systems such as Lyons and Tetsutani's facial control of musical parameters (2001). In their system a video image of a performer's face is analysed in real time to extract facial features such as mouth shape which are then mapped to musical filters such as wah-wah for guitar effects. Whilst this is still not producing music itself, it is providing finer grained control of musical parameters.

3.2 Individual Virtual Instruments

Instruments such as the EpipE (Cannon et al., 2003) attempt to replicate conventional instruments' physicality with the aid of physical input devices and computer modelling. Instruments have also been modelled solely within a computer - virtual instruments. Marshall et al. (2002) developed a virtual Bodhran, the Vodhran. This instrument exists entirely in a computer model and is controlled through 6 degree of freedom input device (x, y, and z axes, as well as pitch, yaw, and roll) whose input data is transformed using a model of a Bodhran's physical properties to produce sound. This differentiates it from augmented instruments in which the input devices attempt to replicate the conventional form factor. Whilst it reduces the problems of capturing players' input through physical input devices such as air flow analysers cf. (Rovan et al., 1997), it also reduces the physical intimacy with the instrument which may lead to changes in the forms of music produced. Indeed, this lack of physicality and the current technological constraints of virtual environments might account for the small number of virtual musical instruments developed. Pressing (Pressing, 1997) highlights the coarse level of control and tactile feedback in current systems along with the high latency response times which would make playing virtual replicas of conventionally designed instruments very difficult indeed.

3.3 Individual New Instruments

There is currently a flourishing field of developments in new forms of musical instruments, especially those intended for individual use. Such developments are exemplified by the majority of work discussed in conferences such as the New Interfaces for Musical Expression series started as a CHI workshop in 2001 (Poupyrev, Lyons, Fels, & Blaine, 2001)(Poupyrev et al., 2001). Such systems aim to provide some novel deterministic mapping from user input to sound output; typically they differ in the form of user input supported (see (Cook, 2001) for a survey of various approaches). Novel individual musical interfaces have the widest range of interaction style so we will outline the various approaches to interaction with musical devices here rather than returning to them in each following section.

There is often a fine line between new instruments and augmented instruments as illustrated by Bernard's work on experimental controllers for music (Bernard, 2002). In his work he explores the boundaries between sound and visual production and produces instruments such as the 'skitar' illustrated in figure 5 - a four stringed electro-acoustic ski which bears physical similarities to a conventional double bass, as does Huott's 'Ski' (2002), or the MIDI-hoover which has physical similarities to a sitar. Similarly, Jordá's QWERTYcaster (2002) attaches a conventional computer keyboard with a joystick and mouse to

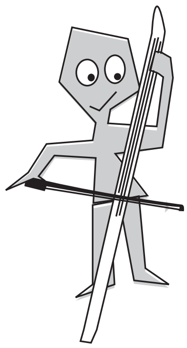

Figure 5: Skitar

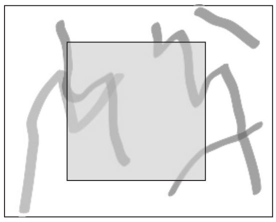

Figure 6: Yellowtail

a guitar shaped piece of wood to allow the performer to 'play the rock guitar hero role'. However, such instruments in no way attempt to replicate conventional instruments, rather they use their visual language to impart some expectation to the devices.

With a clean design slate for new instruments we see forms of user input ranging from standard mouse and keyboard input to video input and physiological sensors. Levin's work on painterly interfaces (2000) epitomises the use of standard input devices as it maps mouse, graphic tablet, and keyboard input to rich visual and auditory displays. In such a system the speed, direction, and position of mouse movements are mapped to sound parameters and to produce visual representations of the music being created. For instance, YelloTail (ibid.) is a reactive paint system; when a user draws a line it is animated by the system as a wiggly worm which travels in a linear or circular manner depending on user specification. As illustrated in figure 6, a square in the middle of the display area is used to generate the sound; when a worm is within the box its pixels are used to generate sound based on the worms' pixels' intensity and position. Levin's work is particularly interesting in terms of the dimensions considered here as it provides deterministic mapping of input to sound based on the squiggles produced, but the squiggles are animated and the sounds producing repeated. In this way it starts to straddle the boundaries of sequencing systems discussed later, but as there is no real facility for describing sounds in the future (as sequencers do), we consider more an instrument which happens to have variable sustain on motifs produced.

Conventional keyboards and mice provide a restricted range of input com-

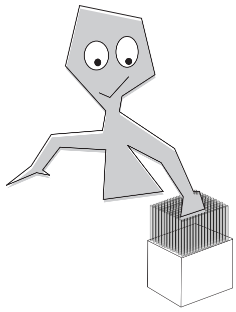

Figure 7: Matrix controller

pared to the finesse of conventional instruments. Capturing human gestures such as hand and body movement opens up the possibility of more rich and expressive control of music (Tarabella & Bertini, 2000). Many systems have been developed using a wide range of motion detecting sensors whose values are mapped to audio parameters. At the simplest level systems such as Beat-Catch (Rydberg & Sandsjö, 2002) which use applied force measurements from one device to control a musical parameter - in this case, the emphasis of beats in a computer generated rhythm. More richly, the Musical Playpen (Weinberg, 1999) consists of a 5' x 5' playpen filled with 400 plastic balls and 4 balls containing piezo-electric accelerometers hidden in the corners. When a child moves within the playpen the balls move around and cause signals to be generated by the four sensors - the acceleration measured is mapped to pitch of note on an Indian-rag scale, and percussive instrument. The interesting aspect of such an approach is that it subverts the traditional mapping of acceleration, or strength of beating which typically relates to volume. A larger scale approach is exemplified by the Electric Circus (Coady, 2002) whose inputs are a 3 x 3 grid of floor switches on which players jump to provide input to the system which is mapped to sound and visuals. Whereas a finer grained approach is taken by the MATRIX (Overholt, 2001) illustrated in figure 7 which is made up of a 12 x 12 grid of spring loaded rods whose vertical positions are measured and used as input to sound production. The form factor is such that an adult's hand just about covers the input surface and so the instrument provides a form of haptic hand control of parameters of sound production ranging from fine grained control of audio waveforms to each rod representing an individual musical instrument.

Video cameras are also used to provide a form of input to new musical instruments as discussed by Tarabella and Bertini (2000). Such approaches typically process the video signal of a camera pointed at a person and map it to audio, with a possible transformation of the video for visual feedback. Hashida et al.'s I-trace (2004) system uses a video camera to capture the position of people within a specified space and then maps these positions to sounds based on a grid of notes in the key of G distributed over the space. Animations around the person are projected onto the floor to give idea of the system state, and to provide a trace of their interaction which is essentially the music being played. Several users can be present in the space at the same time, though there is no explicit support for collaboration.

Finer grained control is exemplified by approaches such as Vogt et al.'s Tongue 'n' Groove musical controller (2002) which takes ultrasound video of a performers mouth and maps it to sound production parameters, so creating an input for new forms of musical instrument. Instruments developed in this way include the 'Tongue-SPASM' which maps tongue height to positions in a computer modelled resonating tube to produce sound continuously controlled by the shape of the player's tongue.

Further away from conventional computer and musical instrument input style we find devices using physiological sensors measuring features such as breath and heart rate. Nagashima (2003) provides an overview of such developments and their use in deterministic control of audio and visual output. For example, their 'breath sensor' converts the expansion of a human's chest during performance into MIDI data in real time which can then be used to somehow transform the performer's voice (acting as an augmented instrument in our classification). Their MiniBioMuse III, on the other hand, measures the electrical activity of muscles on a performer's body using a 16 channel electromyogram attached to the body. The resultant values are used in real time to generate sounds based on the amount of tension in muscles of different parts of the performer's body. Indeed, some of Nagashima's work includes bio-feedback from the system to the performer in the form of electric pulses applied directly to parts of the performer's body in order to control their body - where other performers are involved (e.g. DJs providing input to control the bio-feedback) we would consider this a form of new instrument for co-located groups (see later).

In a wider temporal dimension, the Audiopad (Patten, Recht, & Ishii, 2002) provides an augmented reality interface for musical performance involving prerecorded samples and transformations of such samples. An augmented reality interface essentially involves projecting some computer based display into the user's physical space, and allowing the user to manipulate physical objects within that space which are monitored and interpreted by the system. In the case of Audiopad illustrated in figure 8, a table is augmented with a projected display and a variety of physical control devices are tracked using RF tags. These 'pucks' can be assigned sets of musical samples which can then be transformed through audio effects determined by the puck's position and orientation. The underlying music production system is Ableton Live (Ableton, AG), but the augmented display allows for richer interaction with the system through two handed manipulation of variables as opposed to mouse based control. Moreover, the physical nature of the display provides for some aspect of visual performance when interacting with Audiopad. As with Levin's work discussed previously (2000), Audiopad starts to blur the deterministic boundary as loops are played continuously and can be manipulated in real time, but we do not consider it a sequencer as samples are not scheduled in the future, instead they are started in real time and then manipulated in real time. It is worth noting that all pucks are tracked, it could be possible for more than one person to interact with Audiopad at one time which would make for an interesting collaborative performance. However, explicit support for such interaction has not yet been considered in the design.

Finally, as with virtual versions of conventional instruments, there are also examples of new instruments which solely exist in the virtual world - using input devices such as data gloves and 6 degrees of freedom trackers to manipulate instruments modelled within the computer. For example, Mulder and Fels (1998)

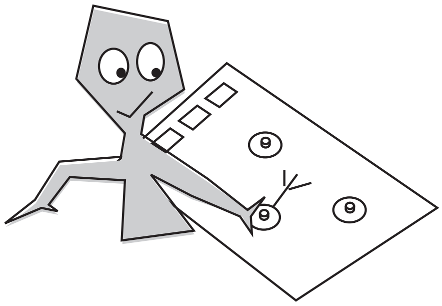

Figure 8: Audiopad

introduced virtual rubber balloons and rubber sheets as ways of interacting with sound. Features of the virtual shapes such as average length of the rubber sheet and average curvature were then mapped to sound production features such as flange and frequency modulation. These mappings are arbitrary and, as they mention, development of more intuitive guidelines for mapping sound parameters would be useful for further development, though they did feel that the intuitive physical characteristcs of the sheet and balloon helped players learn to make sound with them.

3.4 Group Co-Present Interfaces

Interfaces which explicitly support more than one player controlling music production parameters whilst co-located all have multiple inputs which are controlled by different people and may control different audio parameters. The simplest of such systems is exemplified by the MusiCocktail (Mazalek & Jehan, 2000) which allows players to control which loops of music are played concurrently by a central music production system via interactive bottles presenting different loops. Musical production systems in this section all share the design feature of supporting multiple players co-ordinating musical production in some computer supported way. Projects such as the Meta-Orchestra (Bongers & Impett, 2001) explore the networking and infrastructure requirements of linking several computer based musical instruments together to co-ordinate their deterministic music production. Such approaches differ to collections of individual instruments which are not interlinked such as the Brain Opera (Back, 1997)(Orth, 1997) as it provides users with input to a joint music production coordinated by the system rather than by virtue of the acoustic space (as happens with a conventional ensemble of musicians). Similarly, Coady's Electric Circus (2002) which comprises a large floor based input system could be used by several players at the same time, sharing the same interface, but the design itself does not explicitly support group interaction in either its underlying technology or the interface itself.

More intimate group interaction is exemplified by the two player interfaces of Tooka (Fels & Vogt, 2002)(Fels, Kaastra, Takahashi, & McCaig, 2004) and 2hearts (McCaig & Fels, 2002). In the 2hearts system players' heartbeats are monitored and used to control not only the tempo of the piece overall, but also the timbre of instruments. This provides a level of individual and group

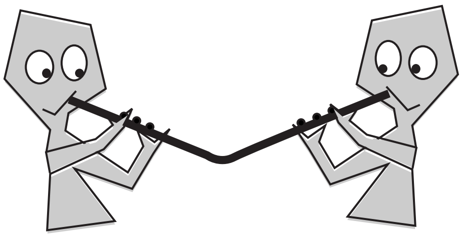

Figure 9: Tooka

music interaction which is not seen in approaches such as the MusiCocktail. Greater control is provided to players of the Tooka - a two player musical instrument which takes input from finger activated buttons and measurements of pressure within a tube physically shared between players. Not only is the pressure in the tube a collaborative product, but the fingering of each player is collaborative, and the bend jointly applied to the tube also provides input to the sound production. Figure 9 illustrates a Tooka with 3 buttons per player and a bend sensor in the middle of the instrument. This, then, is a truly collaborative musical instrument which relies on physical proximity to function. Other approaches foster different musical roles for different players. For example, Squeezables (Weinberg & Gan, 2001) are a set of hand sized squeezable input devices which capture squeezing and pulling gestures of players in order to provide multi-player input to music production (each player typically has one input device). The mapping of squeezables to music production parameters is not uniform - some players play accompaniment squeezables, and typically one plays a melody soloist squeezable. Interrelationships are defined between the players' inputs by the system, e.g. the melody is influenced by the accompaniments, and each accompaniment might control a different aspect of the sound such as features of the synthesizer used. As such we consider the squeezables to be deterministic in their mapping for the most part; some of the balls are less deterministic as the sounds produced is determined in some way by what the others in the group are doing. We return to such non-determinism in other systems later in this survey.

3.5 Group Remote Interfaces

There are no interfaces specifically aimed at players synchronously performing music whilst remotely located. This may be due to bandwidth or technical issues; some semi-synchronous approaches are discussed later. Currently the only support for synchronously performing music with others whilst not in the same physical space is provided by video-conferencing or remote immersion systems. These do not allow users to produce music themselves, but instead focus on connecting remote locations which high quality live audio and video. They key for successful co-performance is low latency between participants; typically with delays under 30ms (Schuett, 2002) which is difficult to achieve over long distances. Recent developments in networking technology have lead to latencies below 100ms (Sawchuk, Chew, Zimmermann, Papadopoulos, & Kyriakakis,

2003) which make co-performance at a distance possible. Once the technology matures and becomes accessible it will be interesting to see how novel interfaces for music could be developed. For example, and networked version of the Tooka (Fels & Vogt, 2002)(Fels et al., 2004) would be feasible and would be a very novel instrument to play when the players are not co-located.

4 Sequencing: Deterministic Mapping and Time

The interfaces recounted in the previous section were instruments with deterministic mapping; they provided an immediate auditory response to player input by mapping input to some auditory signal. In this section we consider interfaces which reduce the level of determinism of the mapping by supporting manipulation of audio over a temporal dimension - the stringing together of audio for editing, storage, and production. We are reducing the level of determinism in such a situation because we do not necessarily know how the whole piece will sound when finished, and the sequence may be edited several times before it is performed.

As discussed in the introduction, storage, editing, and performance of technologically mediated music has a long history reflecting human kind's fascination with machines being able to replay stored composition. Nowadays such systems are popularly referred to as sequencers, and are typically concerned with sequencing MIDI data either by storing copies of the sequence of MIDI data from a MIDI instrument such as a keyboard, or manually entering the MIDI values somehow (Penfold, 1992). Such sequences of MIDI values can then be edited using computer software and played back through MIDI synthesizers. This paper takes a broader view of a sequencer as some piece of software which stores and allows modification to a sequence of values which can be deterministically mapped to audio signals. In our expanded definition values might refer to pre-recorded audio (referred to as samples) as well as MIDI values.

As with the previous section, this section first outlines the interfaces designed for individual use, then group interfaces are categorised in terms of collaborators' physical proximity. There is a much smaller range of interaction styles for sequencers compared to instruments, though many of the forms of interaction exemplified by the instruments already discussed would be eminently suitable as musical input devices for them.

4.1 Individual Sequencing

One approach to the problem of laying out musical sequences is to provide direct input and manipulation of musical representations such as the orthochronic notation (Read, 1969) used in conventional Western music notation. Many commercial systems are available to support such an activity (some early examples are given in (Cope, 1993)), typically using a desktop computer and mouse to enter and edit notes in a score which can then be played in a variety of ways by the computer. Such systems benefit a classically trained composer who is able to understand the notation which is essentially incomprehensible to novices. Leaving aside the design of the notation for the moment, one of the major stumbling blocks for computer based orthochronic notation editors is the difference in affordance between interacting with a pen and paper to produce the score and the

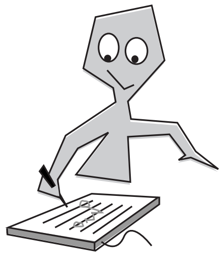

Figure 10: Musical Notepad

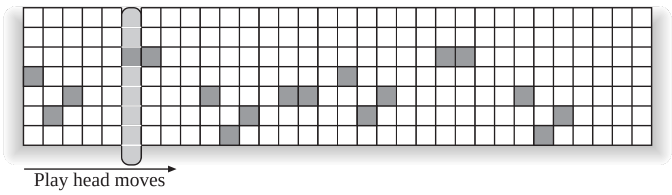

Figure 11: Typical layout of a musical timeline

mouse based interface. In a mouse based interface notes are typically dragged, or placed onto staff lines which, given the resolution of the display being used can be a difficult task. Forsberg et al.'s Musical Notepad (1998) attempts to address this issue by supporting pen based rather than mouse based input of notes. Specifically, allowing composers to draw notes using gestures similar to conventional music notation rather than selecting them from a menu of possibilities as illustrated in figure 10. This brings the music composition process closer to the conventional pen and paper approach whilst supporting editing and play back of the piece. However, it is not much use to users who have little or no understanding of the notation used.

Approaches which support composition and performance of pieces by users who are not classically trained typically abstract away from the detail of the music in some way. One way is to create more simplistic notations such as a musical timeline which typically lays the sequence of notes in a loop from left to right as would be seen in a device such as a music sequencer (e.g. figure 11). Notes are indicated by coloured squares with pitch represented on the vertical axis. When the loop is played, there is some indication of the current position of the 'play head' in the loop such as highlighted note(s), or a line drawn at the appropriate position (the light grey bar in the example). The play head moves along the sequence of notes from left to right. When the end of the loop is reached at the far right, the loop starts again at the beginning. Such representations emphasize the sequential nature of the notes in the loop.

Another approach is to support composition at a meta level - stringing together sequences of sequences of samples rather than individual notes. Typically this allows users to quickly create engaging pieces. Commercial products such as Ableton's Live provide interfaces which allow users to string together overlapping sequence of samples stored in their computer. Playback of the composition can also be manipulated in realtime to create a more interactive performance e.g. by applying transformations to samples, changing the volume of samples, and so on. In order to retain a similar level of musical granularity within this paper we will limit discussion of sequencers to the level of sequencing samples rather than whole songs such as Pauws et al.'s new interfaces for jukeboxes in the home (2000).

Commercial individual sequencers include products such as Steinberg's Cubase (tm) which provide comprehensive and complex recording and editing facilities. Such systems typically allow for recording of audio onto tracks which can then be replayed in parallel, edited, and transformed in a number of ways such as adding filters or effects. In this way they provide a composition tool based on external input and computer based arrangement and transformation, but there is typically no support for sound generation (as opposed to sound recording) within the system. Other approaches record, store, and allow manipulation of MIDI signals from other devices. Whilst a virtual MIDI instrument may be connected to such sequencers, the model behind these sequences is still that it simply records input information (in this case MIDI data) and allows editing and transformation on the recorded information.

Some approaches such as Hyperscore (Farbood & Jennings, 2004) provide both meta level and sequence level composition. The typical motivation of such an approach is to provide novice musicians with an environment in which they can easily move between different musical concepts and foci without formal musical training. In Hyperscore short sequences, referred to as motives, are created using the usual simple sequencer grid style layout with time on the x axis and pitch on the y axis as illustrated in figure 12a. The novel feature of their interface is that each motif is associated with a unique colour. This colour is then used to sketch out the whole piece in a meta composition window with x as the temporal axis and the y axis controlling aspects of the motives such as chord and key changes as illustrated in the compositional sketches in figure 12b. Reported usage indicates that such and approach can increase the level of interest in the musical creation process (as players are no longer stuck with just short sequences) and that there is an increase in complexity and richness of music produced. Whilst the system is explicitly concerned with composition, it would be interesting to explore how the sketching interface could be used in real time performance and improvisation, and as mentioned in their work, how it could be used in conjunction with other MIDI devices. It would also be pertinent to explore how people collaborate when using such a multi level compositional tool, and what cues would be needed to support such interaction.

Unlike the instruments discussed in the previous section, there is a distinct lack of interfaces for composition which are based on novel input techniques. Whilst approaches such as Hyperscore could be developed to take input other MIDI devices, and so possibly any of the novel interfaces previously discussed, there are very few systems which actually try to tackle this problem. For example, Gunther et al.'s work (2002)on composing for music and the sense of touch involved the use of vibrotactile transducers attached to listener's bodies during the playback of pre-composed pieces, however, the composition of the sound and touch pieces was still accomplished using standard MIDI sequencing software rather than considering how to create a touch oriented composition system.

A notable exception to the lack of novel interaction is the Augmented Com-

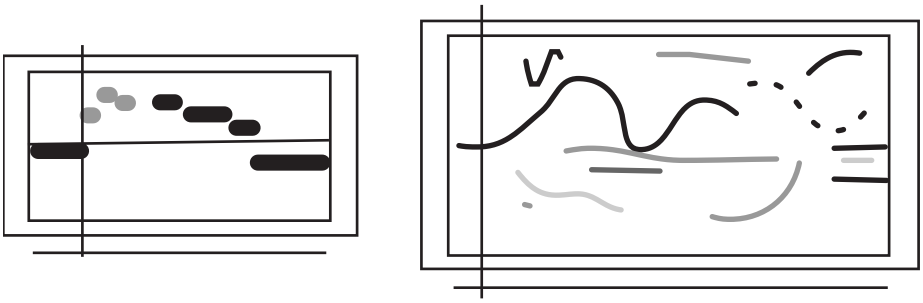

Figure 12: a) Hyperscore motives b) Hyperscore sketches

poser project (Berry, Makino, Hikawa, & Suzuki, 2003) which concerns itself with the use of augmented reality in the support of novices' composition. In one of their approaches - the Music Table - paper cards with symbols printed on them are placed on a table which the system views using a video camera. This video image is displayed on a large screen with various animations added to the cards to indicate the state of the composition. At the sequence composition level 'note cards' are used to indicate notes with the x axis of the tabletop representing time, and the y axis pitch. Once a sequence has been created a 'copy card' is used to place the sequence into a 'phrase card' which then holds the sequence for future playback and editing. These phrase cards can then be put on the table to create meta level compositions and to interact with in real time e.g. changing instruments. Each of the cards is animated in the large display - louder notes are indicated by larger creatures on the cards, sequences are indicated by marching creatures on the phrase cards. In this way they provide a novel interface which is fun, easy to use, and has enough depth of composition to retain interest. The augmented nature of the interface raises questions about how other instruments or input devices could be integrated into the scheme, and how collaboration could be supported at a distance.

4.2 Group Sequencing

Whilst performance of music is intuitively a group effort, composition is often thought of as the province of the solitary composer. However, creativity is often fostered and encouraged when working in groups, and is an essential part of music making (Weinberg & Gan, 2001); this section outlines approaches to support for group sequencing when participants are either co-located or physically distant.

In keeping with our view on individual sequencing, we are concerned here with sequencing of notes or samples, not whole pieces as exemplified by Flytrap (Crossen, Budzik, & Hammond, 2002) which selects songs based on which users are present in a room, thus acting as a form of collaborative jukebox.

The simplest way to support collaboration in sequencing would be to simply enlarge individual instruments. Hankins et al.'s COOL (2002) illustrates such an approach - they provide a large surface on which participants can place objects which are sensed by the computer and used to sequence a group of samples (much like a large version of an Augmented composer). However, there is no explicit technological or interface support for collaboration between players - rather they rely on the size of the physical space to support collaboration. An approach such as Block Jam (Newton-Dunn, Nakano, & Gibson, 2003) neatly illustrates this distinction. Block Jam itself is essentially an individual sequencer comprising physical blocks which can be joined together creating networks of rhythm and sample generating devices. However, developments of the approach include We Jam (ibid) which allows groups of blocks to communicate with each other through a local network using MIDI, and moreover, explorations of their use were carried out with asymmetric dynamics assigned to different people's sets of blocks thus creating a sense of identity in the collaborative experience. Such a development provides a shared music space in the device but does not support any explicit human interaction over and above the music being produced so players would have to rely on their usual co-presence to support interaction.

Wang and Cook (2004) illustrate an interesting approach to co-present group sequencing with their on-the-fly programming concept. In such an approach players create pieces of code which manipulate data to create audio (so, in a sense determining what will happen in the future through the code). The data used as input can come from wave generators, or from other pieces of code, so creating a group composition space. However, there is no explicit support for communication and co-ordination between players (though they could subvert the use of comments in the code to communicate), so players have to be co-located to co-ordinate their actions. In the rest of this section we outline some approaches to supporting group sequencing where composers only communicate through the system itself.

4.3 Remote Group Sequencing

Asynchronous remote group sequencing is technologically the simplest form and is exemplified by systems which allow user to share compositions as they are being worked on. FMOL (Faust Music On Line) illustrates such an approach by providing a central server in which compositions are stored as they are worked on by remote composers. Indeed, the visualisation of music embodied in FMOL was such that it provided an interesting visualisation for co-present performace (Jordá, 2002). Such a framework can be developed to support more synchronous interaction such as jamming, or improvisation with pieces (Wüst & Jordá, 2001). WebDrum (Burk, 2000) is a classic example of such semi-synchronous collaboration support - this system comprises a central server which shares contributions to a short loop of music with users across the internet as illustrated in figure 13. The loop is replicated at each client meaning that there is, in effect, semi-synchronous interaction with the shared loop. Rudimentary co-ordination and ownership of instruments in the shared composition is provided, but the physical remoteness of participants can make composition difficult. Manzolli et al. (2002) generalised such an approach by providing Java support for constructing such collaborative instruments.

It is salient that most work on remote interfaces have drawn design inspiration from conventional approaches such as sequencers. Work including MetaTone (Leach, 2001), and Daisyphone (Bryan-Kinns, 2004) has explored different directions by exploring the features of human communication needed to support more engaging remote composition in such a semi-synchronous framework where we no longer have our co-present physical cues such as sight, gesture,

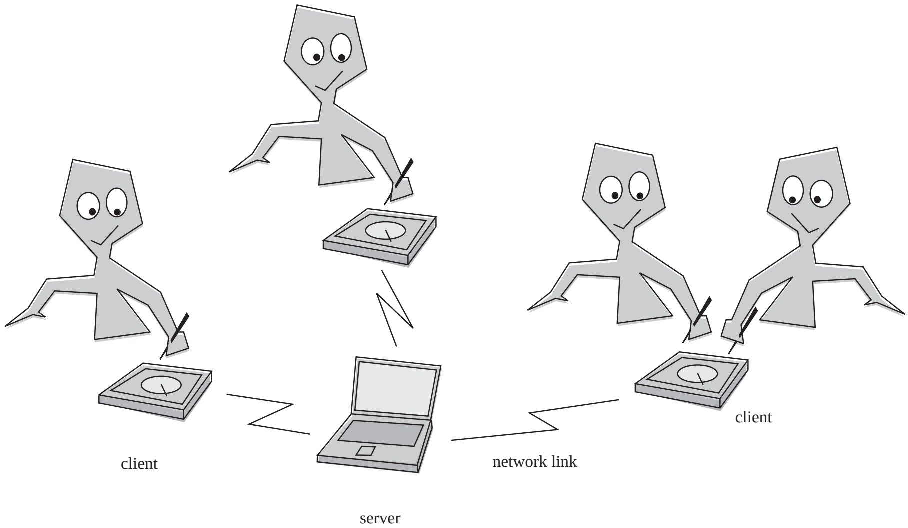

Figure 13: Client server configuration

and breath. Features investigated include the ability to refer to specific parts of compositions, the sense of identity in remote performances, and the nature of mutually-modifiable pieces. The focus of these projects is much more on supporting the group music process rather than the underlying data sharing mechanisms themselves (which are based on Burk's work (2000)). As such they provided richer representation of participants as well as the ability to annotate specific parts of music and localise interaction to points of musical interest. For example, Daisyphone provides a shared space in which music and graphical annotation occur with equal value, and instead of a linear sequencer structure, circular representations are used to reinforce the notion of looping music. Furthermore, contributions can persistent over time, though experiments have been undertaken to investigate the effects of persistence on group musical expression - this will be a key issue for future developments. Such work needs to be developed further in a wider range of settings in order to inform the design of truly engaging new musical instruments from a perspective which is not rooted in conventional instruments. Moreover, understanding the nature of human communication in performance will help to re-integrate audience with performers in music by allowing more fluid boundaries to exist between participants. Only in this way will we realise the true potential of using computers in support of musical expression.

5 Regulating Performance: Non-Deterministic Mapping of Input using Musical Norms

Whilst sequencers embody some level of non-determinism as sequences are composed and edited to be performed in the future, there is still an underlying de-

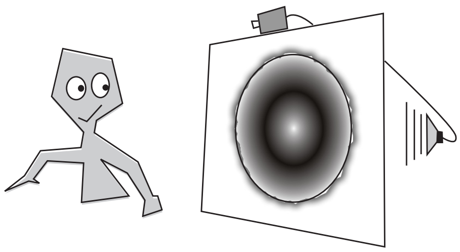

Figure 14: MusiKalscope

terministic mapping from the input value (e.g. a note) to the audio produced in the future (e.g. the sound of the note). In this section we consider interfaces whose underlying mapping of input to audio production is non deterministic because players' input is in some way manipulated to fit with musical rules embodied in the system.

5.1 Individual Regulation

The Colouring-in-Piano CiP; (Oshima, Nishimoto, Miyagawa, & Shirosaki, 2002) provides a salient example of a musical interface which regulates a players' performance in some way. Such a system allows players to perform set pieces (e.g. "Grande Polonaise Brillante Op. 22" by F. Chopin) using a MIDI piano keyboard connected to the computer running the system. When performers play a note on the piano keyboard, it is the expression put into the key press that is mapped, not which note was pressed - the note to be played is determined by a stored MIDI file of the piece. Thus a performer indicates that a new note should occur and provides some expression for that note. However, the piece does not deviate from the stored file in terms of the notes played which, as the system was designed in part to support learning, may actually make mastery of the instrument difficult. Moreover, it is not clear what level of musical expression can actually be achieved if performers are not in control of their own note selection.

Similarly, the MusiKalscope (Fels, Nishimoto, & Mase, 1998) embodies an engine which restricts players contributions to notes that would fit in a Be-Bop improvisation, though they use a novel input device - the Iamascope (Fels, 2000). The Iamascope takes as input a live video stream of the player and projects it onto a large screen in front of the player in a kaleidoscopic fashion - a pie shaped slice of the scene is reflected around a circle as illustrated in figure 14. This kaleidoscopic image is tinted according to the type of notes being produced by the musical part of the system - the RhyMe which produces sounds based on user input from polhemus trackers in the player's hands and a model of jazz for the song being performed. In this way players play along to a recorded song, but are constrained in which notes they play - they essentially select from a list of appropriate notes at the particular point in time. Whilst this makes playing along to the piece easy for novices, it is difficult to see how appropriate such an approach would be for more expert musicians.

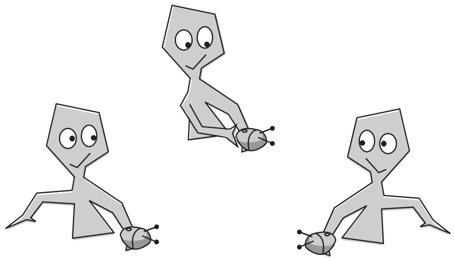

Figure 15: Beatbugs

5.2 Group Regulation

Interfaces designed for groups which embody some form of regulation in the mapping of input to audio production tend to focus on ensuring that collaborators' contributions are in keeping with each other in terms of rhythm or harmony - this emphasises the collaborative and social aspects of music performance and aims to increase involvement in the process by reducing the emphasis on musical prowess.

Prime examples of group regulation embodied in a system are the Fireflies (Weinberg, Lackner, & Jay, 2000) and Beat Bugs (Jennings, 2003) in which co-located players manipulate small hand held devices to create and manipulate short rhythmic loops. Input is through two buttons as illustrated in figure 15 which are used to enter accented and non-accented percussion sounds, stop playback, and share loops. The beats themselves are quantized (stored to the nearest beat), so making synchronisation of multiple players' patterns easier - this quantization is the indeterminancy that differentiates such and approach from group sequencing discussed in previous sections. When the player presses the trade button their patterns are exchanged and can be activated by the player in synchrony with their own pattern so helping to foster some understanding of rhythmic composition. The sharing of patterns between players differentiates this approach from developments such as Jam-o-Drum (Blaine & Perkis, 2000) which provide call-and-response interaction, but with the system itself doing the calling (defining a new sequence or rhythm), rather than other participants, or Jam-o-Whirl (Blaine & Forlines, 2002) in which players chose sets of samples to be played in synchrony and harmony with others' samples. It is interesting to note that such approaches have typically focussed on children as the intended players. Beat Bugs have been used in 'Toy Orchestra' performances which combine children, Beat Bugs, and a professional orchestra to create a novel musical and educational experience. In this way they explore both the pedagogic potential of novel musical instruments as well as the empowering effects of being able to quickly produce and engage in musical performance. Such directions need to be explored further in order for music to regain its central role as a social medium in our Western culture.

6 Musical Partnership

With non-deterministic mapping a player cannot predict the result of providing input to a system. A range of systems exhibit such behaviour ranging from generative music production where a user sets values of parameters which then are used to generate a sequence of music, to systems which respond to, and can accompany a performer's own musical production as a partner. These are all systems in which the music essentially has a life of its own within the system, and the player's input somehow shapes this music over time.

The technology behind a system which can generate its own music ranges from simplistic random number generation to those which involve some form of Artificial Intelligence; (Roads, 1985) provides an overview of AI techniques used in tackling computer based music issues such as developing assistants for composition, generative modelling of music, and understanding features of musical sound. In this paper we categorise interfaces in terms of their employment of AI techniques to generate audio based in some way on user input i.e. their role in providing non-deterministic mapping from input to audio output.

6.1 Partnership with the Individual

The Singing Tree (Oliver, Yu, & Metois, 1997) which was part of the Brain Opera (Back, 1997)(Orth, 1997) exemplifies the use of random number generators involved in mapping of user input in order to generate music which has 'definite trends without being overtly deterministic'. Their approach analyses a user's vocal input as captured by a microphone for changes in pitch (size and speed of change). These changes are then used to select instruments and musical pieces to be performed based on their probability of inclusion and an element of randomness. So, the user controls the instrument in an indirect, non-deterministic manner through the probabilistic mapping of their input, and yet, from observations, users are able to learn, and to some extent control the instrument.

A more sophistic approach to non-deterministic mapping is to somehow model a users input and generate musical sequences which are in keeping. Pachet and Addessi (2004) describe the Continuator which uses Markov-models of melodies contributed by a user via a MIDI piano keyboard to probabilistically generate its own tune. Such a model typically contains a graph of contributed notes where arcs are assigned the probability of transition from one note to another. Accounts of Continuator's use have been positive with both accomplished musicians and novices such as young children. Indeed, one professional musician described it as an 'amplifying mirror'. Such positive responses highlight the possibilities of the technique though it is worth remembering that in its current form they are limit to call-and-response style interaction where a user plays some notes and then the system responds with its tune.

More advanced use of AI techniques attempt to accompany a performer as they play music. These typically involve either an explicit model of the musical genre being played, or a mechanism to identify the structure and then accompany the performer within that model. Explicit models of musical genre are also exemplified by systems which compose improvisational lines without user input such as Horowitz's (1995) modelling of Louis Artmstrong's style using an approach based on Minsky's work on Society of Mind (1986) where intelligence is formed from the coming together of many unintelligent units or agents. In this case the agents represent musical concepts such as features of notes (articulation, colour, harmony etc.), or low-level rhythm, and the knowledge lines connecting many musical concepts represent different interpretations of the same piece. The interpretations of the piece are then used to generate improvisations based on the set goals of the system by a process of spreading activation through the network. However, whilst such an approach does show promise, it is restricted in terms of the genre it can improvise in, and the level of interactivity supported. A more flexible approach is embodied in Thom's BoB system (2000) which learns soloing styles using an unsupervised algorithm to examine the pitch histograms of solos. Such learning can then be used to probabilistically generate solos, but again there is no user interaction with the system.

Other approaches such as the use of genetic algorithms to identify pieces of music evolved over time (Papadopoulos & Wiggins, 1998) tend not to be intended for real-time performance. These concentrate on how pieces can be algorithmically evolved, and how systems can determine which pieces best match requirements such as jazz styled music. In contrast, generative approaches to music making as espoused by composer Brian Eno do provide for real-time interaction. Unlike jazz improvisation techniques where the system attempts to play along with the performer, generative approaches continually produce evolving algorithmic music whose constraints such as musical cohesion, key, and rhythm can be changed by the player, (Yu, 1996). In this way they are more like non-deterministic individual instruments than a musical partner.

6.2 Partnership with a Group

Finally we come to non-deterministic mapping of groups' input to musical output. This is typified by systems which listen to performers (e.g. through microphones or MIDI keyboards) and then somehow accompany the group. In such a mode the computer system comes the closest to a musical partner, and the furthest away from being a musical instrument.

Walker (1994) illustrates a system which could accompany groups of people (or, indeed other systems). His approach is based on models of musical style to inform music analysis and generation, and conversation analysis to understand when and what is appropriate to play (e.g. solos or accompaniment). However, the complexity of attempting real-time group improvisation is evident in the simplistic nature of the models employed which rely on hard coded models of improvisation structure (these become unwieldy for more than a couple of people) and simple rules for generating melody based on using libraries of riffs, random notes selected from the current scale, or copying from others' solos. However, it does provide a concrete step towards the aim of computer systems which can play music with us.

Feldmaier et al.'s work (2002) on dance interfaces take partnership interfaces in a different direction by analysing the movement of dancers in a performance space. Movement is captured using RF tags attached to performers and is analysed for features such as rate of motion. These are then used to control the production of computer generated music. Reflection on its use suggested that whilst it is an intriguing system, there is a lack of depth to pieces generated - the system responded too directly to dancers and so it was difficult to develop a sense

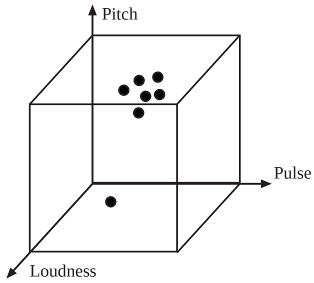

Figure 16: Swarms

of progression or tension through the performance. Possibly employing models of musical structure could help to develop a system which responds to performers whilst having its own sense of direction and purpose in the performance.

Blackwell's swarm approach (2003) takes a approach direction by modelling the flocking nature of insects and using the positions of these moving points as input to sound generators. Figure 16 illustrates a swarm of objects in a 3D space whose axes are mapped to pitch of notes, loudness, and pulse (to synchronize notes produced). The swarms themselves produce coherent music through the constantly self-organising positions of the constituent points. Furthermore, swarms can be attracted to targets which could be defined by humans or by other swarms. In this way the system allow for performance with groups of people and computers. In rejecting explicit models of musical style such as jazz improvisation, Blackwell's approach is flexible and responsive, and moreover, he suggests it generates many new musical ideas. However, it should be noted that the swarms can be controlled through user created 'scripts' which constrain the kinds of music produced. These scripts, then, implicitly embody some model of the music to be produced such as keys and scales.

7 Future directions

In this survey we explored the field of computer interfaces for music production with a range of determinism in transformation, and a range of support for collaboration. In this section we draw out key themes for future directions in the area evident from the work surveyed. Our view on future directions in the field stems from a telling observation: it is clear that whilst there are many, many novel instruments being developed, these are rarely used by anyone other than the creator, and seldom beyond the prototypical stage. Some, such as Levin's painterly interfaces (2000) are used in public performances by the creator. Others, such as Beat Bugs (Jennings, 2003) are used collaboratively by others in public performance. However, such systems are exceptions to the rule. The most widely used computer music systems are based on conventional metaphors such as Steinberg's Cubase (tm) computer based simulation of a recording studio, or Abelton's Live which is a development on the theme of conventional Djing. This raises the question of why we are in this situation. It could be a product of limited access to new instruments typically developed as unique items in research labs, or because they are difficult to learn, or difficult to reconstruct, or maybe they simply do not sound musical to a large enough group of people. Below are some possible directions to address such issues.

7.1 Expanding Access

Whilst we have seen an increase in the number of commercial sequencers on computers such as Apple Computer Inc.'s GarageBand (tm), and on mobile devices such as Sony Corp.'s MusicDJ (tm) for their mobile telephones, these are based on fairly conventional models of music interaction and do not involve new input devices. For new input devices to flourish we will need to develop standard components which can be mass produced and which use standard communication protocols such as MIDI. It will probably not be possible or desirable in the short term to mass produce novel instruments. Rather, individual components could be produced with some means of easily interconnecting them. This relates to Bongers and Harris' (2002) structured approach to instrument design, but will also need to encompass graphic techniques for connecting inputs and devices such as PD (Puckette, 1996). This requires mapping out the range of input devices in order to understand the different modailities and haptics of input which could be usefully employed. Moreover, once we start to see some standardisation of understanding interfaces, we will be in a better position to start to explore the development of hybrid approaches such as mixing video analysis with 3D position sensors, or mouth control with a conventional sequencer which would currently require building and/ or integrating systems from scratch.

Hunt et al. (2002) touched upon the idea of developing guidelines for mapping from input devices to audio parameters. These might include guidelines such as when to use a slider instead of a mouse, or how to best create multiple mappings between input devices. Indeed, they suggest a two layer design model relating concrete input parameters to abstract conceptual models. Such approaches need to be developed further to create richer and more intriguing musical interfaces from a systematic design process. In this way we will start to see some of the rigour typified by other approaches such as graphic design or human-computer-interaction applied and exploited in the field.

7.2 Increasing Take-Up

Conventional musical instruments are hard to play. They take years of practice and training to master. Once mastered, however, they can be used to produce rich and engaging music for both player and listeners. A key theme in the research surveyed here is how to make new instruments which are easy to master yet provide for a great depth of expression. One possible reason for the scarcity of use of new instruments is that they are either easy to use and so lack expressive depth, or that they take a long time to master which has mitigated their use. (Jordá, 2004) presents a framework for thinking about the learnability as well as expressiveness of instruments and similarly argues that we should be designing instruments which are easy to learn, and yet have expressive depth contrasting a kazoo (easy to learn, low expressive depth) with a violin (very hard to learn, but great expressive depth). A similar view is taken by Wessel and Wright (2001) who are interested in developing systems with a 'low entry fee [and] with no ceiling on virtuosity'. One approach to addressing this issue may be use of artificial intelligence techniques to model, understand, and react to player's ability. This would expand the use of artificial intelligence in music beyond the typical use as an improvisational partner e.g. (Walker, 1994). We could also see this approach as increasing the level of non-determinism in mapping from input to sound production. For example, novice players would be provided with direct mapping from input to sound, and as they developed, more scope for non-determinism of mapping could be introduced such as the system generating repetitions, or playing on previous themes.

7.3 Coming Together

Music making is fundamentally a social process, yet it is interesting to note that the predominant forms of interaction with new, computer based musical instruments are oriented to individual use. That is, there is no explicit support for interaction with others over and above physical proximity. This comes at a time when computers are increasingly interconnected through local networks, wireless connectivity, and the pervasive use to the internet. We suggest two primary reasons for this below.

First, we need to further explore standards for supporting remote synchronous collaborative music making such as Open Sound Control (Wright et al., 2003). Such approaches have not yet gained as wide acceptance as MIDI which is only really suitable for co-located communication of music data, and neither protocol provides any coherent scope for communicating non-music data used to co-ordinate activities such as gesture, verbal interaction, gaze between participants, and spatial and proximal awareness. Until such standards exist it will be difficult to design and build musical instruments which are intended to connect with others to support music making when participants are not in the same space.

Second, computer based musical instruments are still being constructed from an individualistic point of view without taking into account how and why they will be used. For example, Audiopad (Patten et al., 2002) supports an individual in manipulating samples. If there were several people playing Audiopad(s) at the same time there would be no way of co-ordinating the activity other than through happening to be physically co-located even though interconnectivity is essentially part of the underlying computer system. This follows from conventional models of musical instrument design where it is assumed that people will interact in the same space which is not necessarily the case in today's networked society. This is compounded by a lack of understanding of human communication which leads to naïve development of group instruments. Some exceptions include Beat Bugs (Jennings, 2003) which support sharing of rhythms between co-located players, but does not have any explicit support for collaboration. Jam-O-World (Blaine & Forlines, 2002) does embody call-response patterns in order to co-ordinate joint action, but such structures are simplistic compared to the richness of human-human interaction such as conversation. In order to address this issue we need to explore what it means to make music from a collaborative point of view and use our understandings to inform the development of more socially oriented musical instruments which can take advantage of the range of interaction now afforded to us.

8 Summary

Devices with which to make music somehow transform our actions into sounds. In this paper we surveyed current computer based approaches in terms of the level of determinism of mapping from action to sound, and, given the social nature of music making, we considered the explicit support for collaboration built into the devices. Current research tends to focus on individual instruments with fairly deterministic mappings from action to sound. In order to develop musical instruments beyond the conventional models we need to explore what it means to collaborate when making music, and how to use such understandings to build new, more engaging forms of musical instruments.

9 References

Back, M. J. (1997). Sound design for brain opera's mind forest: Audio for a complex interactive system. *Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques (DIS '97), Amsterdam, The Netherlands* (pp. 23–25). New York, USA: ACM.

Bailey, D. (1992). *Improvisation: Its nature and practise in music.* London, UK: The British Library National Sound Archive.

Bernard, D. (2002, May). Experimental controllers for live electronic music performance (vs. copyright). *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02), Dublin, Ireland.*

Berry, R., Makino, M., Hikawa, N., & Suzuki, M. (2003). The augmented composer project: The music table. *Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '03).* Los Alamitos, CA, USA: IEEE Computer Society.

Blackwell, T. M. (2003, April). Swarm music: Improvised music with multi-swarms. In Gervás, & Colton (Eds.), *Proceedings of 2003 AISB Symposium on Artificial Intelligence and Creativity in Arts and Science, University of Wales, Aberystwyth, United Kingdom* (pp. 41–49).

Blaine, T., & Fels, S. (2003, May). Contexts of collaborative musical experiences. In F. Thibault (Ed.), *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03), Montreal, Canada* (pp. 129–134). McGill University, Montréal, Québec, Canada: Faculty of Music.

Blaine, T., & Forlines, C. (2002, May). Jam-o-world: Evolution of the jam-o-drum multi-player musical controller into the jam-o-whirl gaming interface. *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02), Dublin, Ireland.*

Blaine, T., & Perkis, T. (2000, August). The jam-o-drum interactive music system: A study in interaction design. In J. Karat, & J. Thackara (Eds.), *Proceedings of DIS 2000, Brooklyn, New York* (pp. 165–173). New York, USA: ACM.

Bongers, B., & Harris, Y. (2002, May). A structured instrument design approach: The video-organ. *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02), Dublin, Ireland.*

Bongers, B., & Impett, J. (2001). The meta-orchestra. a narrative: from theory to practice... to theory again. hypermusic and the sighting of sound. Project Report sponsored by EC Connect 1999 scheme and organised by the Dartington International Summer School in Devon, England.

Bowers, J. (2002). Improvising machines. Master's thesis, Masters in Music by Research, University of East Anglia, Norwich, UK.

Bowles, E. A. (1970). Musicke's handmaiden: Or technology in the service of the arts. In H. B. Lincoln (Ed.), *The computer and music* (Chap. 1, pp. 3–23). Cornell University Press.

Bryan-Kinns, N. (2004, August). Daisyphone: The design and impact of a novel environment for remote group music improvisation. In D. Benyon, & P. Moody (Eds.), *Proceedings of DIS 2004, Boston, Mass., USA*. ACM, New York, USA: ACM.

Burk, P. (2000). Jammin' on the web - a new client/server architecture for multi-user musical performance. *Proceedings of International Computer Music Conference (ICMC 2000), Berlin, Germany*.

Cannon, C., Hughes, S., & Modhráin, S. (2003, May). Epipe: Exploration of the uilleann pipes as a potential controller for computer-based music. In F. Thibault (Ed.), *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03), Montreal, Canada* (pp. 3–8). McGill University, Montréal, Québec, Canada: Faculty of Music.

Chadabe, J. (2002, May). The limitations of mapping as a structural descriptive in electronic instruments. *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02), Dublin, Ireland*.

Coady, N. (2002). Interaction design for real-time music creation environments - the electric circus. Master's thesis, MA, University of Westminster, UK.

Cook, P. (2001, April). Principles for designing computer music controllers. *Proceedings of CHI 2001; NIME workshop, Seattle, USA*. ACM.

Cope, D. (1993). *New directions in music*. USA: Brown and Benchmark, sixth edition.

Crossen, A., Budzik, J., & Hammond, K. J. (2002). Flytrap: Intelligent group music recommendation. *Proceedings of Intelligent User Interfaces 2002, San Francisco, CA, USA* (pp. 184–185). New York, USA: ACM.

Crowhurst, N. H. (1971). *Electronic musical instruments*. Tab Books.

Ernst, D. (1977). *The evolution of electronic music*. New York, USA: Schrimer Books, New York, USA.

Farbood, M. M., & Jennings, K. (2004). Hyperscore: A graphical sketchpad for novice composers. *IEEE Emerging Technologies*, 50–54.

Feldmeier, M., Malinowski, M., & Paradiso, J. A. (2002). Large group musical interaction using disposable wireless motion sensors. *Proceedings of 2002 International Computer Music Conference (ICMC 2002), Gothenburg, Sweden* (pp. 83–87). San Francisco, USA: International Computer Music Association.

Fels, S. (2000). Intimacy and embodiment: Implications for art and technology. In S. Ghandeharizadeh, S.-F. Chang, S. Fischer, J. Konstan, & K. Nahrstedt (Eds.), *ACM Multimedia 2000* (pp. 13–16). New York, USA: ACM.

Fels, S., Kaastra, L., Takahashi, S., & McCaig, G. (2004, June). Evolving tooka: from experiment to instrument. In Y. Nagashima (Ed.), *Proceedings of International Conference on New Interfaces for Musical Expression (NIME 04)* (pp. 1–6). Shizuoka University of Art and Culture, Hamamatsu, Japan: Department of Art and Science, Faculty of Design.

Fels, S., Nishimoto, K., & Mase, K. (1998). Musikalscope: A graphical musical instrument. *IEEE Multimedia Magazine*, *5*(3), 26–35.

Fels, S., & Vogt, F. (2002). Tooka: Explorations of two person instruments. *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02), Dublin, Ireland.*

Forsberg, A., Dieterich, M., & Zeleznik, R. (1998, November). The music notepad. In E. Mynatt, & R. J. K. Jacob (Eds.), *Proceedings of 11th annual symposium on*
User Interface Software and Technology (UIST '98). San Francisco, CA, USA (pp. 203–210). New York, USA: ACM.

Gunther, E., Davenport, G., & O'Modhrain, S. (2002). Cutaneous grooves: Composing for the sense of touch. *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02), Dublin, Ireland.*

Hankins, T., Merrill, D., & Robert, J. (2002). Circular optical object locator. *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02), Dublin, Ireland.*

Hashida, T., Kakehi, Y., & Naemura, T. (2004, June). Ensemble system with i-trace. In Y. Nagashima (Ed.), *Proceedings of International Conference on New Interfaces for Musical Expression (NIME 04)* (pp. 215–216). Shizuoka University of Art and Culture, Hamamatsu, Japan: Department of Art and Science, Faculty of Design.

Holmes, T. B. (1985). *Electronic and experimental music.* Charles Scribner's Sons, New York, USA.

Horowitz, D. (1995). Representing musical knowledge in a jazz improvisation system. *International Joint Conference on AI (IJCAI-95) Workshop On Artificial Intelligence And Music.*

Hunt, A., Wanderley, M. M., & Paradis, M. (2002). The importance of parameter mapping in electronic instrument design. *Proceedings of the 2002 Conference on New Instruments for Musical Expression, Dublin, Ireland.*

Huott, R. (2002). An interface for precise musical control. *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02), Dublin, Ireland.*

Jennings, K. (2003). 'toy symphony': An interactional music technology project for children. *Music Education International*, *21*(2), 3–21.

Jordá, S. (2002). Improvising with computers: A personal survey (1989-2001). *Journal of New Music Research*, *31*(1).

Jordá, S. (2004, June). Digital instrument and players: Part i - efficiency and apprenticeship. In Y. Nagashima (Ed.), *Proceedings of International Conference on New Interfaces for Musical Expression (NIME 04)* (pp. 59–63). Shizuoka University of Art and Culture, Hamamatsu, Japan: Department of Art and Science, Faculty of Design.

Leach, J. (2001). Metatone: Shared environment for musical collaboration. Master's thesis, Department of Computer Science, Queen Mary, University of London, London, UK.

Levin, G. (2000). Painterly interfaces for audiovisual performance. Master's thesis, Master of Science in Media Arts and Sciences at the Massachusetts Institute of Technology, MIT, Boston, Mass., USA.

Li, T.-C. (1999). Who or what is making the music: Music creation in a machine age. In E. Edmonds, & L. Candy (Eds.), *Proceedings of the third conference on Creativity & cognition, Loughborough, UK* (pp. 57–62). New York, USA: ACM.

Lyons, M. J., & Tetsutani, N. (2001). Facing the music a facial action controlled musical interface. In J. Jacko, & A. Sears (Eds.), *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '01)* (pp. 309–310). New York, USA: ACM.

Machover, T., & Chung, J. (1989). Hyperinstruments: Musically intelligent and interactive performance and creativity systems. *Proceedings of the International Computer Music Conference (ICMC 1989), Columbus, Ohio, USA.*

Manelis, J. (2002). Adaptive hyperinstruments: Applying evolutionary techniques to sound synthesis and performance. *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02), Dublin, Ireland.*

Manning, P. (1985). *Electronic and computer music.* Claredon Press, Oxford.

Manzolli, J., Costa, M. O., Ramos, F. L., Fornari, J. E., & Sharoni, D. (2002). Solutions for distributed musical instruments on the web. In J. Wladron, & J. Power (Eds.), *Proceedings of the inaugural conference on the Principles and Practice of programming, 2002* (pp. 77–82). New York, USA: ACM.

Marshall, M., Rath, M., & Moynihan, B. (2002). The virtual bodhran - the vodhran. *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02), Dublin, Ireland.*

Matthews, M. (1969). *The technology of computer music.* Cambridge, Massachusetts, USA: The MIT Press.

Mazalek, A., & Jehan, T. (2000, April). Interacting with music in a social setting. *Proceedings of CHI 2000 Conference on Human Factors in Computing Systems extended abstracts, The Hague, NL.* (pp. 255–256). New York, USA: ACM.

McCaig, G., & Fels, S. (2002). Playing on heart-strings: Experiences with the 2hearts system. *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02), Dublin, Ireland.*

MIDI, M. A. (1996). *The complete midi 1.0 detailed specification.* DREAM(ATMEL), Germany: MIDI Manufacturers Association (MMA) European Acting Agent.

Minsky, M. (1986). *The society of mind.* New York, USA: Simon and Schuster.

Moore, G. E. (1965). Cramming more components onto integrated circuits. *Electronics*, *38*(8).

Mulder, A. G. E., & Fels, S. (1998). Sound sculpting: Manipulating sound through virtual sculpting. *Proceedings of Western Computer Graphics Symposium, Whilster, BC, Canada* (pp. 15–23).

Nagashima, Y. (2003, May). Bio-sensing systems and bio-feedback systems for interactive media arts. In F. Thibault (Ed.), *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03), Montreal, Canada* (pp. 48–53). McGill University, Montréal, Québec, Canada: Faculty of Music.

Newton-Dunn, H., Nakano, H., & Gibson, J. (2003, May). Blockjam: A tangible interface for interactive music. In F. Thibault (Ed.), *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03), Montreal, Canada* (pp. 170–177). McGill University, Montréal, Québec, Canada: Faculty of Music.

Oliver, W., Yu, J., & Metois, E. (1997). The singing tree design of an interactive musical interface. In S. Coles (Ed.), *Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques (DIS '97), Amsterdam, The Netherlands* (pp. 261–264). New York, USA: ACM.

Ord-Hume, A. W. J. C. (1980). *Musical box. a history and collector's guide.* London, UK: George Allen and Unwin.

Orio, N., Schnell, N., & Wanderley, M. M. (2001, April). Input devices for musical expression: Borrowing tools from hci. *Proceedings of CHI 2001; NIME workshop, Seattle, USA.* New York, USA: ACM.

Orth, M. (1997, April). Interface to architecture: Integrating technology into the environment in the brain opera. *Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques (DIS '97), Amsterdam, The Netherlands* (pp. 265–275). New York, USA: ACM.

Oshima, C., Nishimoto, K., Miyagawa, Y., & Shirosaki, T. (2002). A concept to facilitate musical expression. *Proceedings of the fourth conference on Creativity & cognition, Loughborough, United Kingdom* (pp. 111–117). New York, USA: ACM.

Overholt, D. (2001). The matrix: A novel controller for musical expression. *Proceedings of CHI 2001; NIME workshop, Seattle, USA.* New York, USA: ACM.

Pachet, F., & Addessi, A. R. (2004). When children reflect on their playing style: Experiments with the continuator and children. *ACM Computers in Entertainment*, *2*(2).

Papadopoulos, G., & Wiggins, G. (1998). A genetic algorithm for the generation of jazz melodies. *Proceedings of the 8th Finnish Artificial Intelligence Conference (STeP'98), Jyväskylä, Finland.* Finland: Finnish Artificial Intelligence Society.

Patten, J., Recht, B., & Ishii, H. (2002). Audiopad: A tag-based interface for musical performance. *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02), Dublin, Ireland.*

Pauws, S., Bouwhuis, D., & Eggen, B. (2000, April). Programming and enjoying music with your eyes closed. *Proceedings of CHI 2000 Conference on Human Factors in Computing Systems, The Hague, NL.* (pp. 376–383). New York, USA: ACM.

Penfold, R. A. (1992). *Computers and music.* UK: PC Publishing.

Pennycook, B. W. (1985). Computer-music interfaces: A survey. *Computing Surveys*, *17*(2), 267–289.

Poupyrev, I., Lyons, M. J., Fels, S., & Blaine, T. (2001). New interfaces for musical expression. In M. M. Tremaine (Ed.), *CHI '01 extended abstracts on Human factors in computing systems, Seattle, USA.* (pp. 491–492). New York, USA: ACM.

Pressing, J. (1997). Some perspectives on performed sound and music in virtual environments. *Presence*, (6), 1–22.

Puckette, M. (1996). Pure data. *Proceedings of International Computer Music Conference (ICMC '96), San Francisco, USA* (pp. 269–272).

Read, G. (1969). *Music notation.* Boston, USA: Crescendo Publishing Co.

Roads, C. (1980). Interview with max mathews. *Computer Music Journal*, *4*(4), 15–22.

Roads, C. (1985). Research in music and artificial intelligence. *Computing Surveys*, *17*(2), 163–190.

Rovan, J. B., Wanderley, M. M., Dubnov, S., & Depalle, P. (1997). Instrumental gestural mapping strategies as expressivity determinants in computer music performance. In A. Camurri (Ed.), *Kansei - The Technology of Emotion Workshop at the AIMI International Workshop, Genova, Itlay.*

Russolo, L. (1913/ 1986). *The art of noises*, Vol. 6 of *Monographs in Musicology.* New York, USA: Pendragon Press.

Rydberg, L., & Sandsjö, J. (2002). Beatcatch: Visual and tactile rhythm box. In O. W. Bertelesen, S. Bodker, & K. Kuutti (Eds.), *Proceedings of the Second Nordic Conference of Human-Computer Interaction (NordiCHI 2002)* (pp. 299–301). New York, USA: ACM.

Sawchuk, A. A., Chew, E., Zimmermann, R., Papadopoulos, C., & Kyriakakis, C. (2003). From remote media immersion to distributed immersive performance. *Proceedings of the 2003 ACM SIGMM workshop on Experiential telepresence (ETP '03)* (pp. 110–120). New York, USA: ACM.

Sawyer, K. (2003). *Group creativity: Music, theater, collaboration.* NJ, USA: Lawrence Erlbaum Associates (LEA).

Schuett, N. (2002). The effects of latency on ensemble performance. Master's thesis, CCRMA Department Of Music, Stanford University, Stanford, California, USA.

Sloboda, J. A., & O'Neill, S. (2001). *Music and emotion: Theory and research*, Chap. Emotions in Everyday Listening to Music, pp. 415–430. UK: Oxford University Press.

Tarabella, L., & Bertini, G. (2000). Giving expression to multimedia performance. *Proceedings of the 2000 ACM workshops on Multimedia* (pp. 35–38). New York, USA: ACM.

Thom, B. (2000). Unsupervised learning and interactive jazz/blues improvisation. In H. Kautz, & B. Porter (Eds.), *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000), Austin, TX, USA.* Menlo Park, California USA: AAAI Press.

Titon, J. T. (1996). *World of music.* New York, USA: Schrimer Books.

Vogt, F., McCaig, G., Ali, M. A., & Fels, S. (2002). Tongue 'n' groove: An ultrasound based music controller. *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02), Dublin, Ireland.*

Walker, W. F. (1994). *A conversation-based framework for musical improvisation.* Graduate College of the University of Illinois at Urbana-Champaign, USA: PhD Thesis in Computer Science.

Wang, G., & Cook, P. R. (2004, June). On-the-fly programming: Using code as an expressive musical instrument. In Y. Nagashima (Ed.), *Proceedings of International Conference on New Interfaces for Musical Expression (NIME 04)* (pp. 138–143). Shizuoka University of Art and Culture, Hamamatsu, Japan: Department of Art and Science, Faculty of Design.

Weinberg, G. (1999). The musical playpen - an immersive digital musical instrument. *Personal and Ubiquitous Computing*, *3*(3).

Weinberg, G., & Gan, S.-L. (2001). The squeezables: Toward an expressive and interdependent multi-player musical instrument. *Computer Music Journal*, *25*(2), 37–45.

Weinberg, G., Lackner, T., & Jay, J. (2000). The musical fireflies - learning about mathematical patterns in music through expression and play. *Proceedings of XII Colloquium on Musical Informatics.* University of Udine, Diploma Universitario per Operatore dei Beni Culturali (Gorizia), Italia: Italian Association for Musical Informatics.

Wessel, D., & Wright, M. (2001). Problems and prospects for intimate musical control of computers. *Proceedings of ACM SIGCHI, CHI '01 Workshop New Interfaces for Musical Expression (NIME'01).* ACM.

Wright, M., Freed, A., & Momeni, A. (2003, May). Open sound control: State of the art 2003. In F. Thibault (Ed.), *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03), Montreal, Canada* (pp. 153–159). McGill University, Montréal, Québec, Canada: Faculty of Music.

Wüst, O., & Jordá, S. (2001). Architectural overview of a system for collaborative music composition over the web. *Proceedings of International Computer Music Conference 2001 Havana, Cuba.*

Young, D. (2002). The hyperbow controller: Real-time dynamics measurement of violin performance. *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02), Dublin, Ireland.*

Young, D., & Essl, G. (2003, May). Hyperpuja: A tibetan singing bowl controller. In F. Thibault (Ed.), *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03), Montreal, Canada* (pp. 9–14). McGill University, Montréal, Québec, Canada: Faculty of Music.

Yu, C. (1996). Computer generated music composition. Master's thesis, Engineering in Electrical Engineering and Computer Science, Massachusetts Institute of Technology, USA.